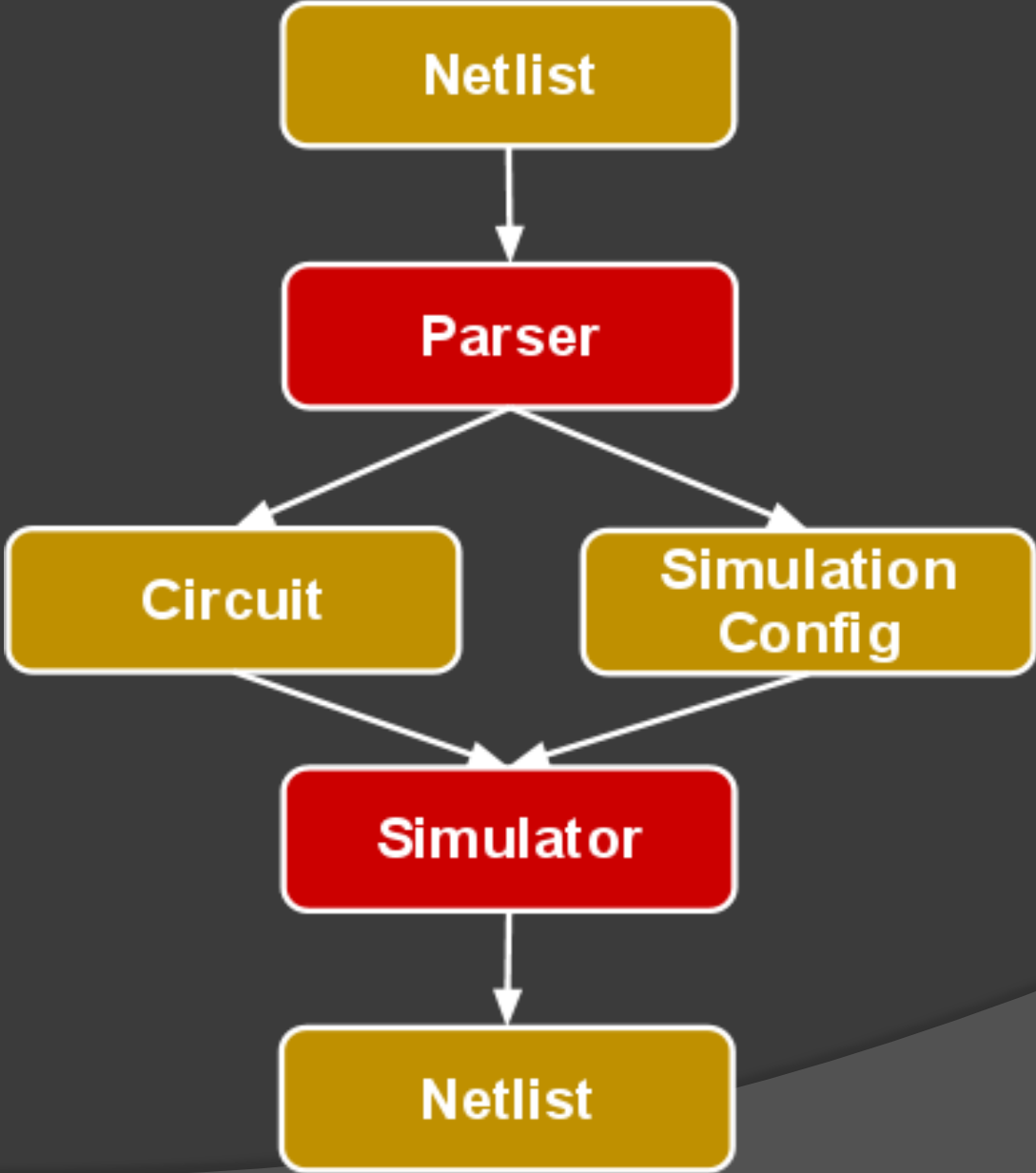


CSPICE

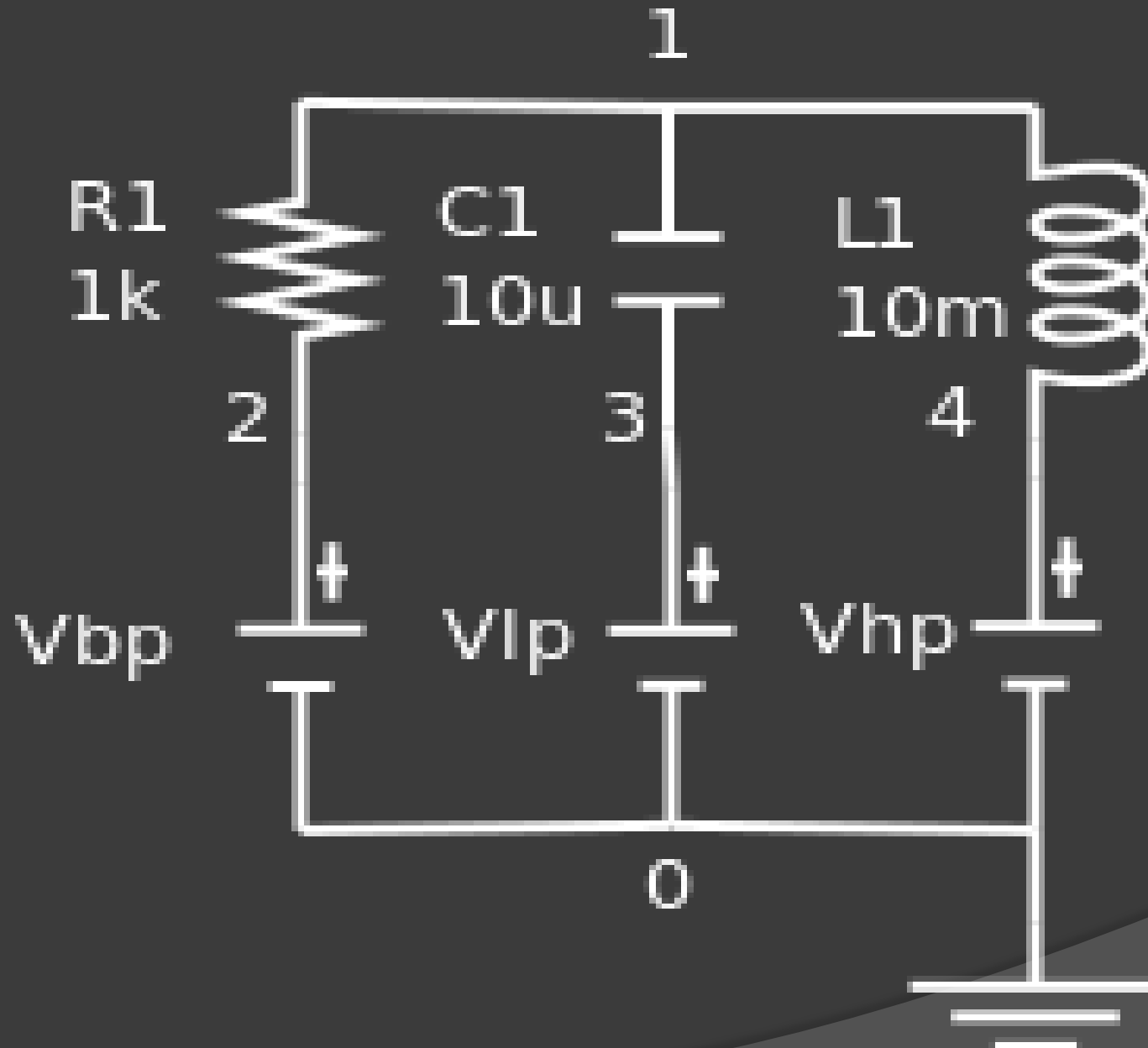
<http://github.com/eggegg/cspice>

陳韋翰
劉彥廷

Structure



An example



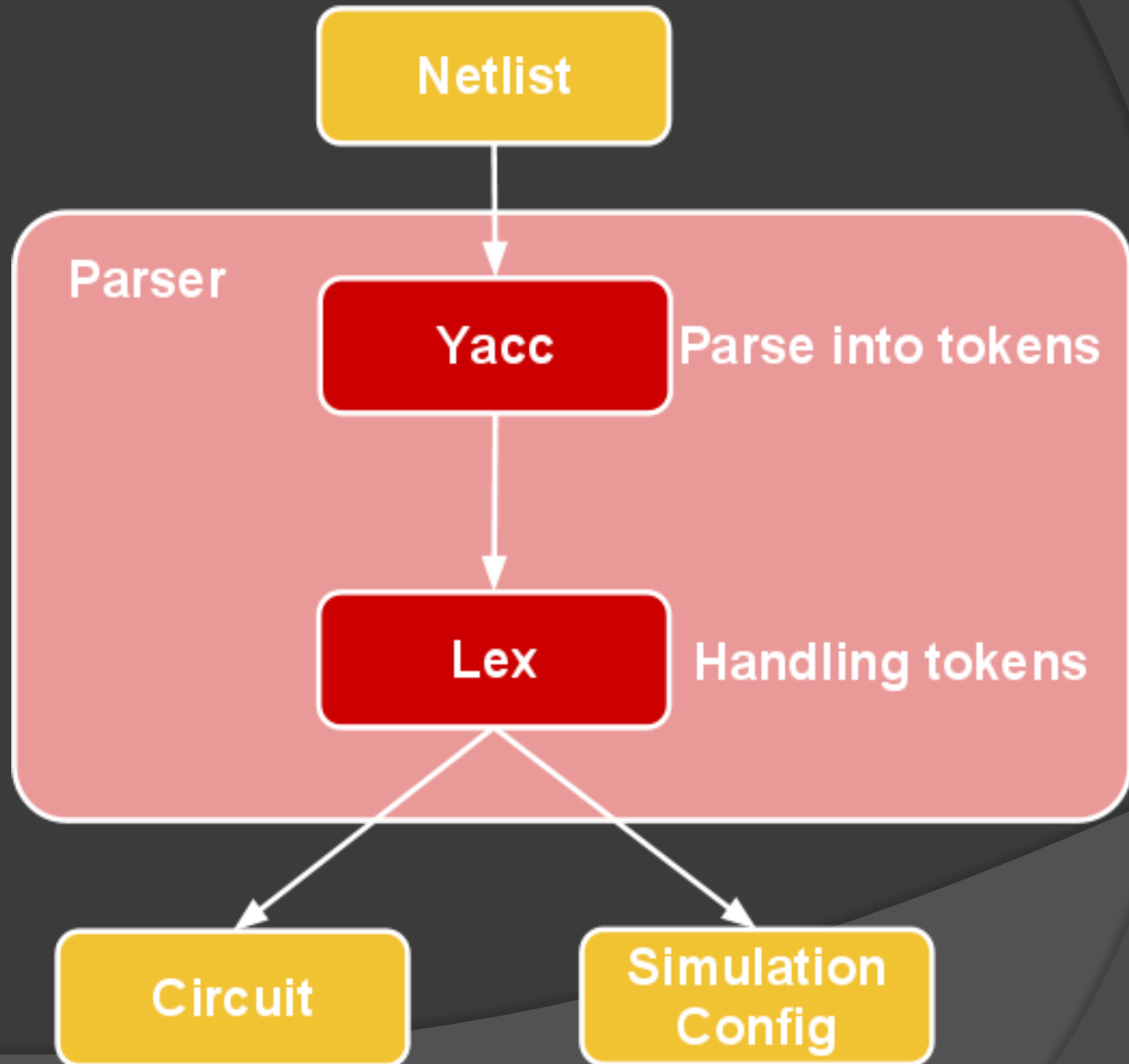
Netlist

```
# This is a sample netlist
OUT 1 0
R1 1 2 1k
C1 1 3 1u
L1 1 4 10m
Vbp 2 0 0 1 # band pass filter
Vlp 3 0 0 0
Vhp 4 0 0 0

FREQ 1 1M 10 Vbp freq_bp.eps
FREQ 1 1M 10 Vlp freq_lp.eps
FREQ 1 1M 10 Vhp freq_hp.eps

TIME 0 1m 1u time.eps
```

Parser



Circuit

==== Circuit Detail ====

[1]

-> [2] R1 1.000e+03

-> [3] C1 1.000e-06

-> [4] L1 1.000e-02

[2]

-> [1] R1 1.000e+03

[3]

-> [1] C1 1.000e-06

[4]

-> [1] L1 1.000e-02

[0]

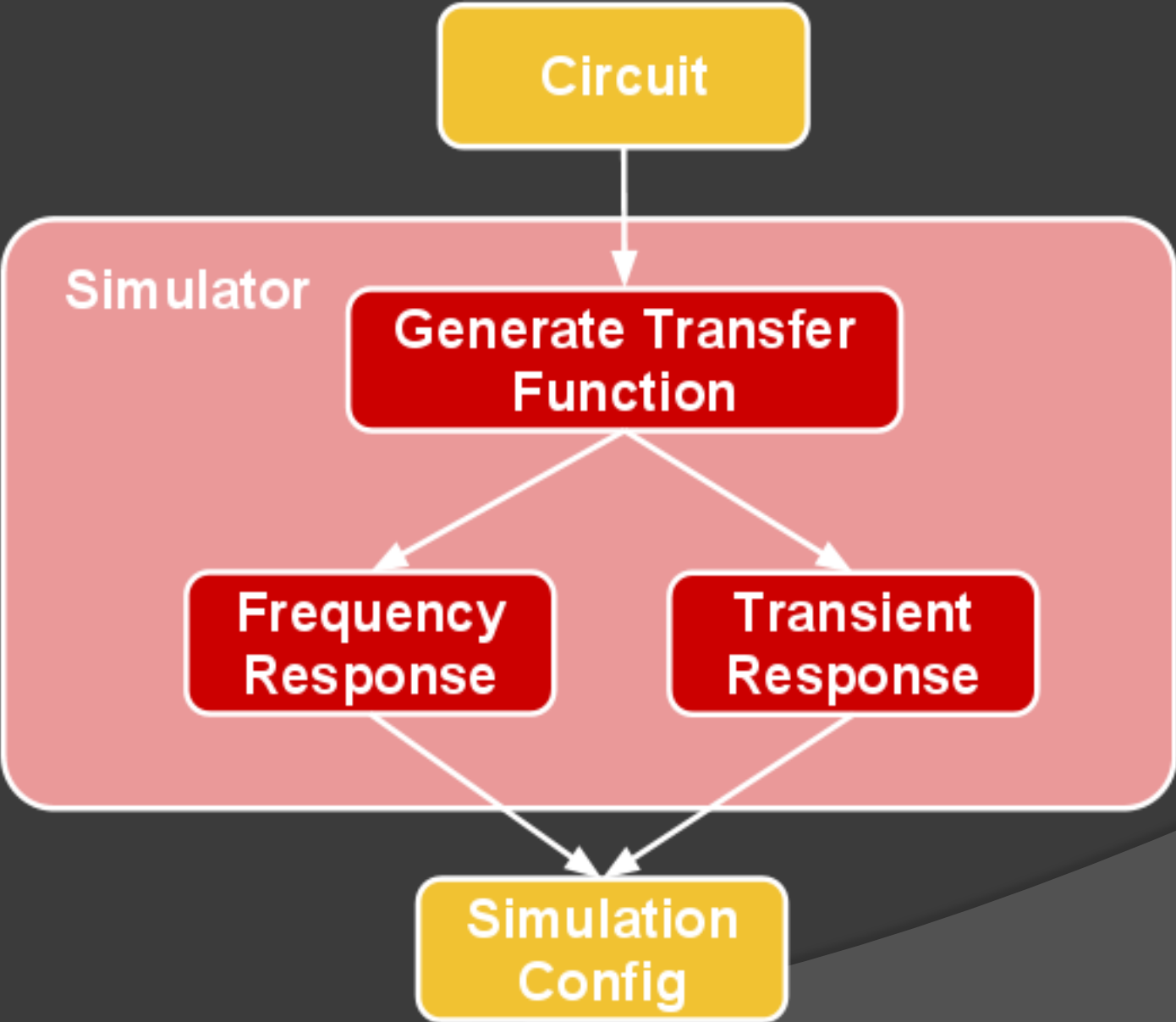
----- Source -----

Vbp [2] -> [0] 0.000e+00 1.000e+00

Vlp [3] -> [0] 0.000e+00 0.000e+00

Vhp [4] -> [0] 0.000e+00 0.000e+00

Simulator



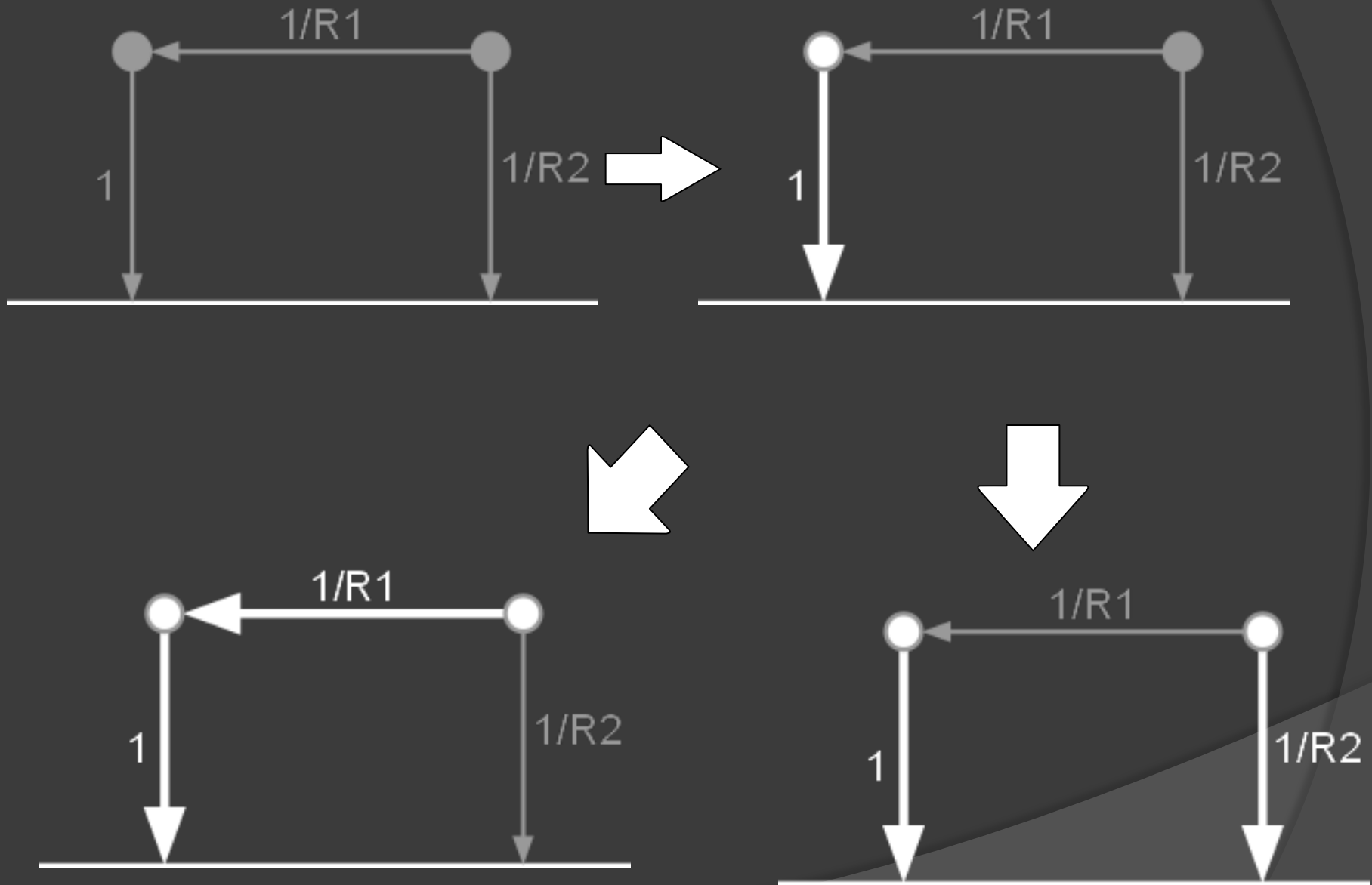
Simulation

- ◎ Frequency Response
 - Must specify an input (voltage/current source)
 - Other sources can be ignored
- ◎ Transient Response
 - Using superposition
 - Shutdown other sources when simulating a single source

Generate Transfer Function

- ⦿ For each VSRC & ISRC
 - Using "input low" as reference node
- ⦿ VSRC:
 - Denominator
 - Add dummy link "1" from "input high" to "input low"
 - Finding spanning trees, sum all the product
 - Numerator
 - Add dummy link "-1" from "input high" to "output high"
 - Add dummy link "1" from "input high" to "output low"
 - Finding spanning trees, sum all the product
- ⦿ ISRC:
 - Denominator
 - Finding spanning trees, sum all the product
 - Numerator
 - Add dummy link "-1" from "input high" to "output low"
 - Add dummy link "1" from "input high" to "output low"
 - Finding spanning trees, sum all the product

Finding spanning trees (DFS)



Generating Data

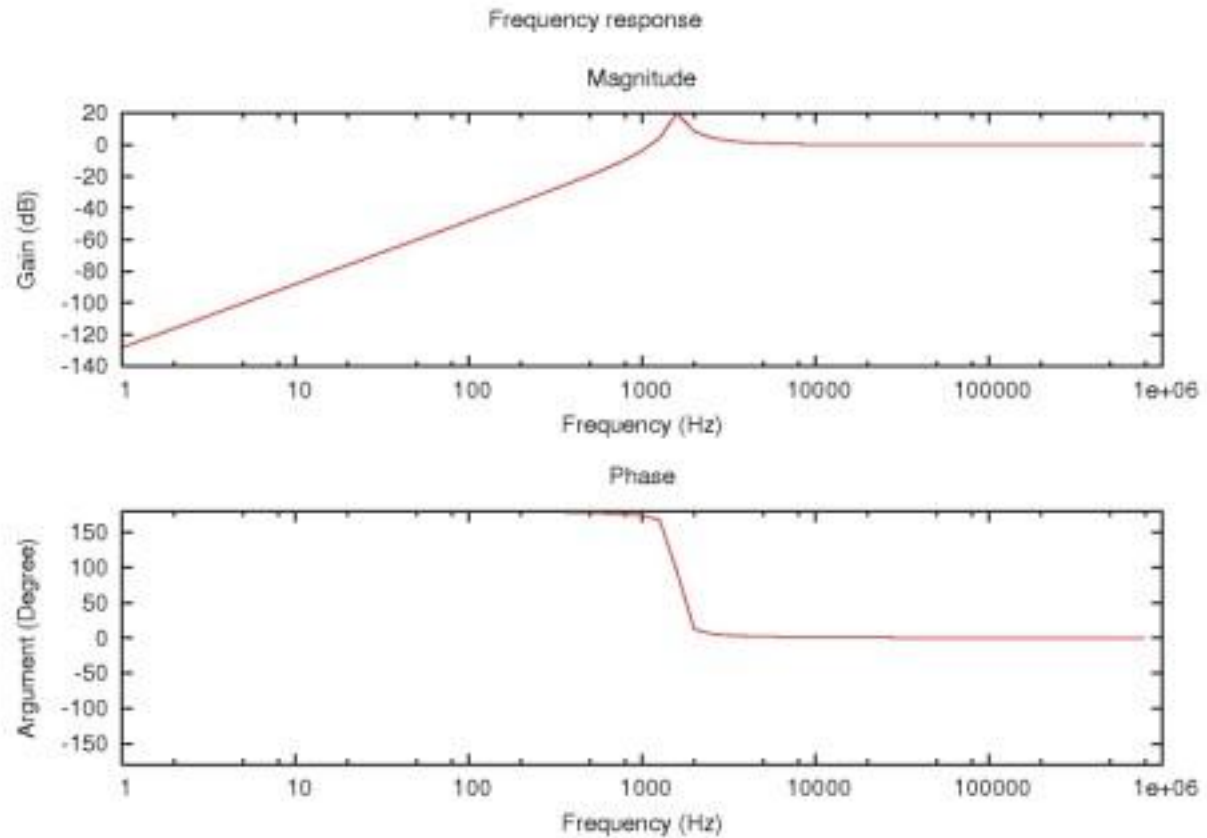
⦿ Frequency Response

- Simply just plot the points of the transfer function

⦿ Transient Response

- Based on Laplace transform
- All input sources are modeled with step function (which is specified in netlist)
- Using trapezoidal rule to perform numerical integration

Frequency response - HP filter



Time response

